Gradient Vanishing

Ki Hyun Kim

nlp.with.deep.learning@gmail.com

Backpropagation with Chain Rule

- Gradient들의 곱셈들로 이루어져 있음
- 입력에 가까운 레이어의 파라미터일수록 곱셈이 늘어남
  - Gradient가 1보다 작을 경우, 좌변은 점점 작아질 것

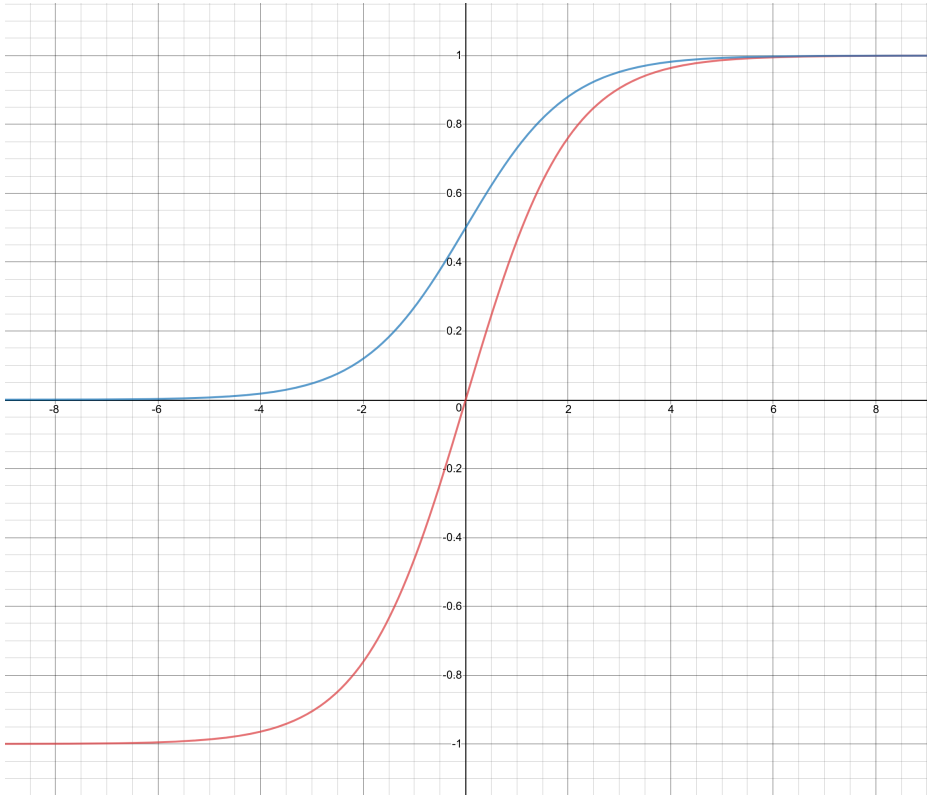
$$\frac{\partial \mathcal{L}}{\partial W_3} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_3}$$

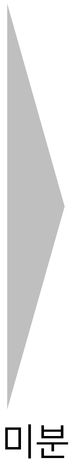
$$\frac{\partial \mathcal{L}}{\partial W_2} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial W_2}$$

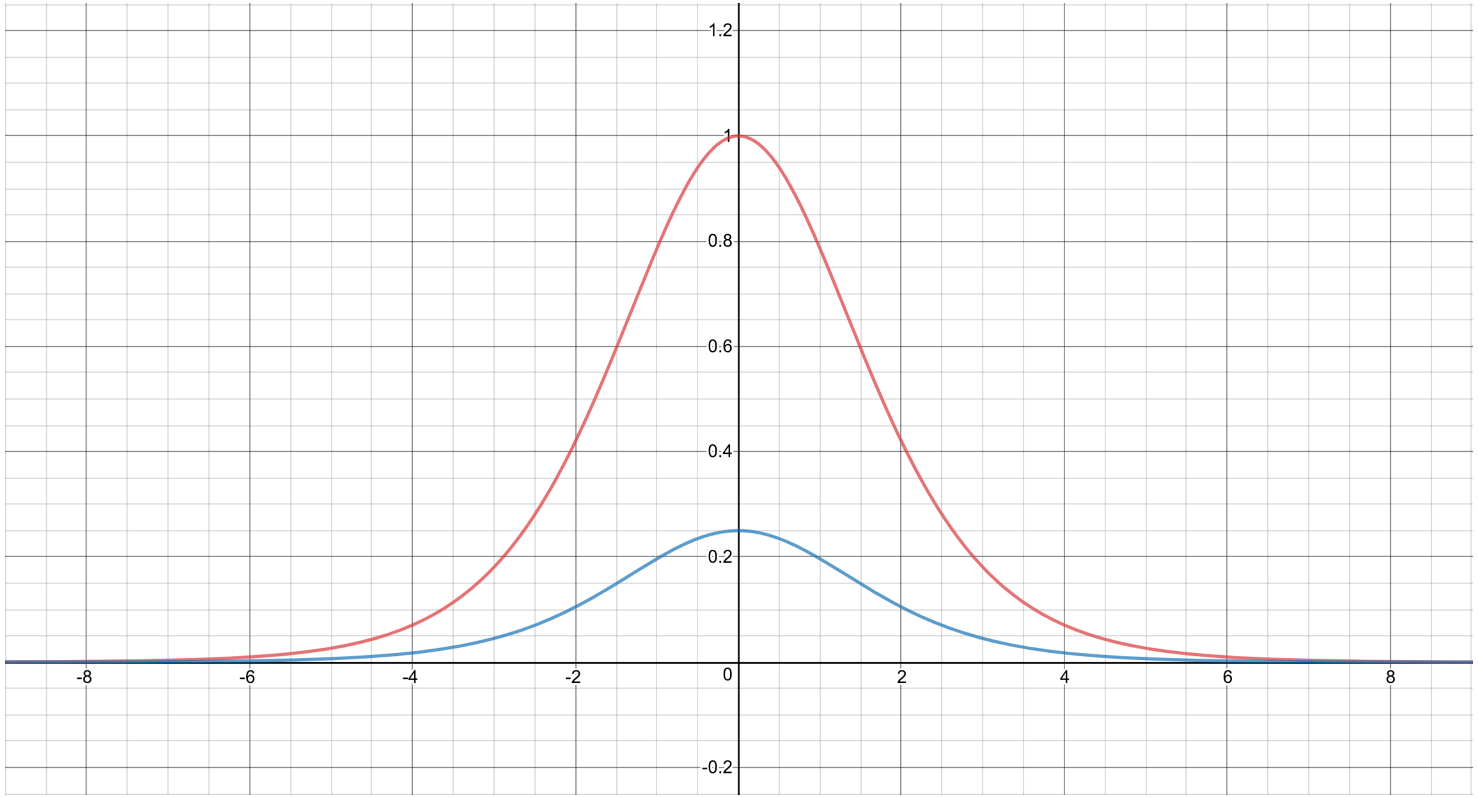
$$\frac{\partial \mathcal{L}}{\partial W_1} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_1}$$

Gradient of Sigmoid & TanH

- 모두 1보다 작거나 같다.







Gradient Vanishing because of Activation Functions

- 깊은 네트워크를 구성하게 되면 점점 gradient가 작아지는 현상
- 따라서 깊은 신경망을 학습하기 어렵게 됨
  - 앞쪽 레이어의 파라미터는 업데이트 되는 크기가 매우 작기 때문

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \|y_i - \hat{y}_i\|_2^2$$

$$\begin{aligned}
\hat{y}_i &= h_{2,i} \cdot W_3 + b_3 \\
h_{2,i} &= \sigma(\tilde{h}_{2,i}) \\
\tilde{h}_{2,i} &= h_{1,i} \cdot W_2 + b_2 \\
h_{1,i} &= \sigma(\tilde{h}_{1,i}) \\
\tilde{h}_{1,i} &= x_i^\intercal \cdot W_1 + b_1
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W_1} &= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_1} \\
&= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial \tilde{h}_2} \cdot \frac{\partial \tilde{h}_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial \tilde{h}_1} \cdot \frac{\partial \tilde{h}_1}{\partial W_1}
\end{aligned}$$

$$\text{where } \frac{\partial h_\ell}{\partial \tilde{h}_\ell} = \frac{\partial \sigma}{\partial \tilde{h}_\ell} < 1.$$